## LIST OF CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application.

Claims 1-4 (cancelled)

5. (Currently amended) A method for scheduling events in a computer processing system, comprising:

identifying ~~multiple~~ queues, each of the ~~multiple~~ queues associated with a corresponding priority, each of the queues including events;

defining a data structure with a root level having a node group, the node group having k number of nodes, each of the k number of nodes sharing a pointer, each of the k number of nodes stored contiguously in memory, wherein the k number is equal to a number of multiple queues;

associating the ~~multiple~~ queues with respective nodes of the data structure;

assigning a value representing the corresponding priority to the respective nodes;

determining a priority between the respective nodes based on respective values representing the corresponding priority to the respective nodes ~~the value~~; and

selecting one of the events ~~multiple queues~~ corresponding to a node having a highest priority for transmission to a processing resource.

6. (Currently amended) The method of claim 5, further comprising:

rescheduling the node having the highest priority ~~one of the multiple queues~~ after selection.

7. (Currently amended) The method of claim 6, wherein the method operation of rescheduling the node having the highest priority ~~one of the multiple queues~~ after selection includes,

determining if the node having the highest priority ~~one of the multiple queues~~ will be empty after selection.

8. (Currently amended) The method of claim 7, further comprising:

if the node having the highest priority ~~one of the multiple queues~~ will be empty after selection, then the method includes,

removing the value representing the corresponding priority from the node having the highest priority ~~respective nodes~~.

9. (Currently amended) The method of claim 7, further comprising:

if the node having the highest priority ~~one of the multiple queues~~ will not be empty after selection, then the method includes,

retaining the value representing the corresponding priority from the node having the highest priority ~~respective nodes~~, thereby enabling rescheduling of the node having the highest priority ~~one of the multiple queues~~ after selection.

10. (previously presented) The method of Claim 5, further comprising:

resolving conflicts between respective nodes assigned a same value by rotating a pointer among the respective nodes assigned the same value.

Claims 11-21 (cancelled)

22. (currently amended) A computer readable medium having program instructions for scheduling events in a computer processing system, comprising:

program instructions for identifying ~~multiple~~ queues, each of the ~~multiple~~ queues associated with a corresponding priority, each of the queues including events;

program instructions for defining a data structure with a root level having a node group, the node group having k number of nodes, each of the k number of nodes sharing a pointer, each of the k number of nodes stored contiguously in memory, wherein the k number is equal to a number of multiple queues;

program instructions for associating the ~~multiple~~ queues with respective nodes of the data structure;

program instructions for assigning a value representing the corresponding priority to the respective nodes;

program instructions for determining a priority between the respective nodes based on respective values representing the corresponding priority to the respective nodes ~~the value~~; and

program instructions for selecting one of the events ~~multiple queues~~ corresponding to a node having a highest priority for transmission to a processing resource.

23. (currently amended) The computer readable medium of claim 22, further comprising:

rescheduling the node having the highest priority ~~one of the multiple queues~~ after selection.

24. (currently amended) The computer readable medium of claim 23, wherein the program instructions for rescheduling the node having the highest priority ~~one of the multiple queues~~ after selection includes,

determining if the node having the highest priority ~~one of the multiple queues~~ will be empty after selection.

25. (currently amended) The computer readable medium of claim 24, further comprising:

if the node having the highest priority ~~one of the multiple queues~~ will be empty after selection, then the computer readable medium includes,

program instructions for removing the value representing the corresponding priority from the node having the highest priority ~~respective nodes~~.

26. (currently amended) The computer readable medium of claim 24, further comprising:

if the node having the highest priority ~~one of the multiple queues~~ will not be empty after selection, then the computer readable medium includes,

program instructions for retaining the value representing the corresponding priority from the node having the highest priority ~~respective nodes~~, thereby enabling rescheduling of the node having the highest priority ~~one of the multiple queues~~ after selection.

27. (original) The computer readable medium of claim 22, further comprising:

program instructions for resolving conflicts between respective nodes assigned a same value by rotating an additional pointer among the respective nodes assigned the same value.